Lecture 14

Bayesian Inference (Part 2)

(Based on slides by **Dr. Tom Rainforth**, HT 2020)

Jiarui Gan

jiarui.gan@cs.ox.ac.uk

This Lecture

- In this lecture we will we show how the foundational methods introduced in the last lecture are not sufficient for inference in high dimensions
- Particular topics:
  - The Curse of Dimensionality
  - Markov Chain Monte Carlo (MCMC)
  - Variational Inference

The Curse of Dimensionality

The Curse of Dimensionality (1)

- The curse of dimensionality is a tendency of modeling and numerical procedures to get **substantially harder as the dimensionality increases**, often at an **exponential rate**
- If not managed properly, it can cripple the performance of inference methods
- It is the main reason the two methods discussed so far, rejection sampling and importance sampling, are in practice only used for very low dimensional problems
- At its core, it stems from an increase of the size (in an informal sense) of a problem as the dimensionality increases

The Curse of Dimensionality (2)

- Imagine calculating an expectation over a discrete distribution of dimension D, where each dimension has K possible values
- The cost of enumerating all the possible combinations scales as K^D (exponential in D); even for modest values for K and D this will be prohibitively large
- The same problem occurs in continuous spaces: think about splitting the space into blocks, we have to reason about all the blocks to reason about the problem

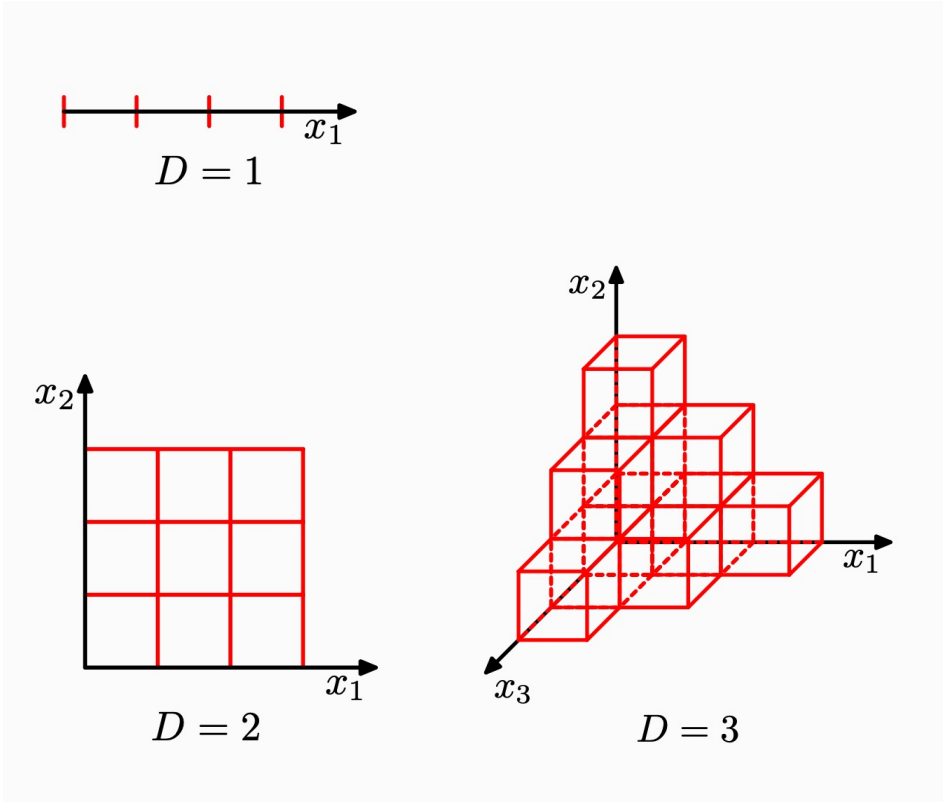
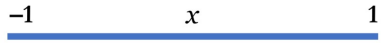


*Image Credit: Bishop, Section 1.4

Example: Rejection Sampling From a Sphere

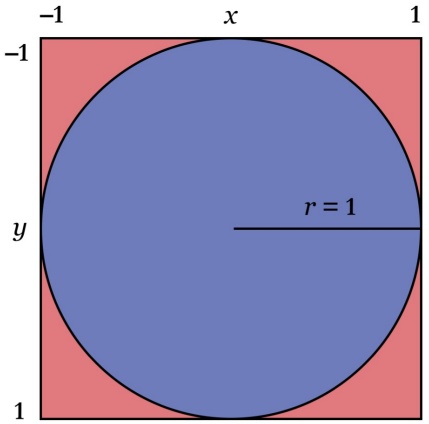
- Consider rejection sampling from a D-dimensional hypersphere with radius r using the tightest possible enclosing box:



1 dimension ($D = 1$)

Sample points x in a segment of length 2

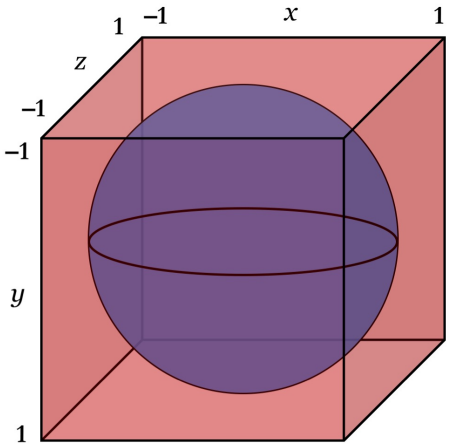
All points are accepted



2 dimensions ($D = 2$)

Sample points (x, y) in a square of side 2

Accept points inside the unit circle



3 dimensions ($D = 3$)

Sample points (x, y, z) in a cube of side 2

Accept points inside the unit sphere

$$P_{\text{Accept}} = \frac{V_{\text{sphere}}}{V_{\text{cube}}} = \left(\frac{\sqrt{\pi}}{2}\right)^{D} \frac{1}{(D/2)!}$$

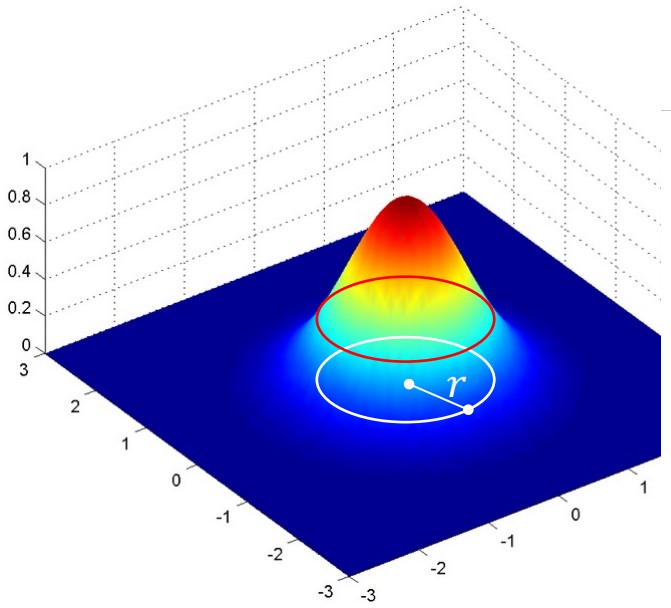
- $D = 2, 10, 20,$ and 100 gives P_{Accept} values of $0.79, 2.5\times10^{-3}, 2.5\times10^{-8},$ and 1.9×10^{-70}, respectively
- OK in low dimensions, but infeasible in higher dimensions

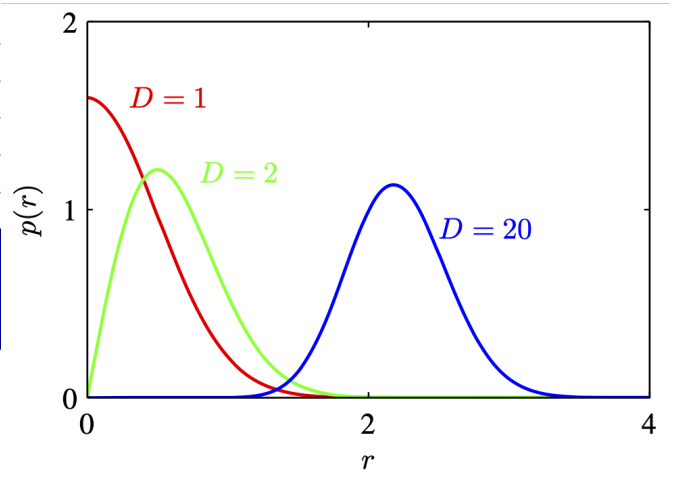
Curse of Dimensionality: Importance/Rejection Sampling

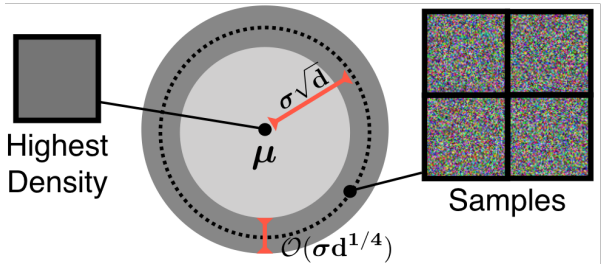
- For both importance sampling and rejection sampling we use a proposal $q(\theta)$ as an approximation of the target $p(\theta|D)$
- As the dimension increases, it quickly becomes much harder to find good approximations
- The performance of both methods typically diminishes exponentially as the dimension increases

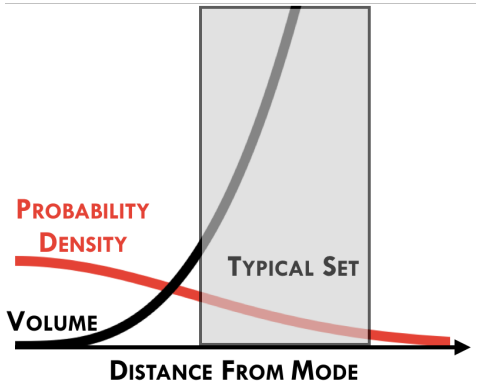
Typical Sets

- Consider representing an isotropic Gaussian in polar coordinates. The marginal density of the radius changes with dimension
- In high dimensions, the posterior mass concentrates in a thin strip **away from the mode** known as the **typical set**
- This means that, not only is the mass concentrated to a small proportion of the space in high dimensions, the geometry of this space can be quite complicated









How Can We Overcome The Curse of Dimensionality?

- As we showed with the typical sets, the area of significant posterior is usually only a small proportion of the overall space
- To overcome the curse, we thus need to use methods which **exploit structure** of the posterior **to only search this small subset** of the overall space
- All successful inference algorithms make some implicit assumptions into the structure and then try to exploit this
  - **MCMC methods** exploit local moves to try and stick within the typical set (thereby also implicitly assuming there are not multiple modes)
  - **Variational methods** assume independences between different dimensions that allow large problems to be broken into multiple smaller problems

Markov Chain Monte Carlo (MCMC)

The Markov Property

- In a Markovian system each state is independent of all the previous states given the last state, i.e.

$$p(\theta_n|\theta_1,\ldots,\theta_{n-1}) = p(\theta_n|\theta_{n-1})$$

  The system transitions based only on its current state.

- **MCMC main idea:** generate every sample θ_n based on the previous sample θ_{n-1}
  - E.g., random walk

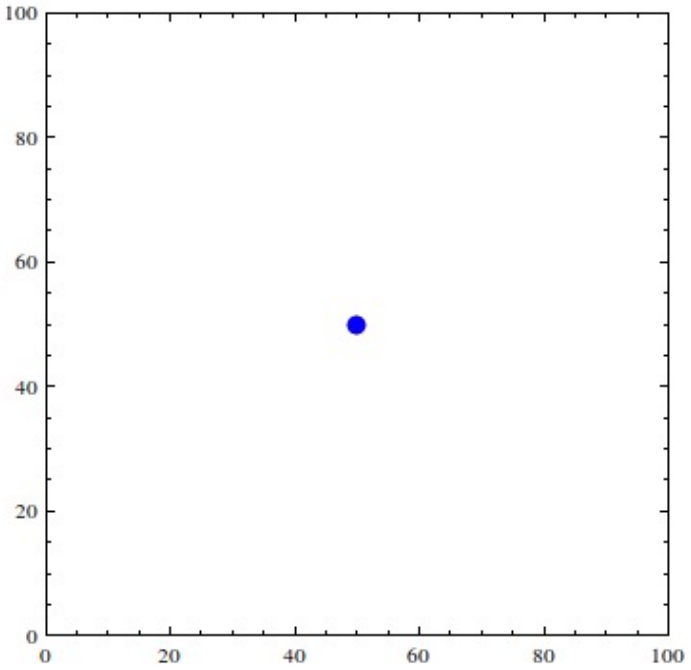


*Image source: https://mathematica.stackexchange.com/questions/111839/random-walk-in-limited-range

Defining a Markov Chain

- All the Markov chains we will deal with are **homogeneous**
- This means that each time step has the same transition dynamic:

$$p(\Theta_{n+1} = \theta' | \Theta_n = \theta) = p(\Theta_n = \theta' | \Theta_{n-1} = \theta)$$

- In such situations, $p(\Theta_{n+1} = \theta' | \theta_n = \theta)$ is typically known as a **transition kernel**, also written as $T(\theta' \leftarrow \theta)$
- The distribution of any homogeneous Markov chain is fully defined by a combination of an **initial distribution** $p(\theta)$ and the **transition kernel** $T(\theta' \leftarrow \theta)$

Markov Chain Monte Carlo (MCMC)

- MCMC methods are one of the most ubiquitous approaches for Bayesian inference and sampling from target distributions more generally
- The key is to construct a valid **Markov chain** that produces sample from the target distribution
- They circumvent the curse of dimensionality by **exploiting local moves**
  - They have a hill–climbing effect until they reach the typical set
  - They then move around the typical set using local moves
  - They tend to fail spectacularly in the presence of multi–modality

Convergence of a Markov Chain (1)

- To use a Markov chain for consistent inference, we need it to be able to produce an **infinite series** of samples that **converge** to our posterior:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=M}^{N} \mathbb{I}(\Theta_n = \theta) = p(\theta|\mathcal{D})$$

  where M is a number of **burn–in** samples that we discard from the start of the chain

- In most cases, a core condition for this to hold is that the distribution of individual samples converge to the target for all possible starting points:

$$\lim_{N\to\infty} p(\Theta_N = \theta'|\Theta_1 = \theta) = p(\theta'|\mathcal{D}) \quad \text{for all } \forall\theta, \theta'$$

Convergence of a Markov Chain (2)

- Ensuring that the chain converges to the target distribution for all possible initializations has two requirements

1. $p(\theta|\mathcal{D})$ must be the **stationary distribution** of the chain, such that if $p(\Theta_n = \theta) = p(\theta|\mathcal{D})$ then $p(\Theta_{n+1} = \theta) = p(\theta|\mathcal{D})$. This is satisfied if:

$$\int T(\theta' \leftarrow \theta) \cdot p(\theta|\mathcal{D}) d\theta = p(\theta'|\mathcal{D})$$

   where we see that the target is invariant to the application of the transition kernel.

2. The Markov chain must be **ergodic**. This means that all possible starting points converge to this distribution.

Ergodicity

- Ergodicity itself has two requirements. The chain must be:
  1. **Irreducible**, i.e., all points with non-zero probability can be reached in a finite number of steps
  2. **Aperiodic**, i.e., no states can only be reached at certain periods of time
- These requirements for these to be satisfied are very mild for commonly used Markov chains, but are beyond the scope of the course
- Optional homework: figure out how we can get the stationary distribution from the transition kernel when θ is discrete
  - Hint: start by defining the transition kernel as a matrix

Detailed Balance

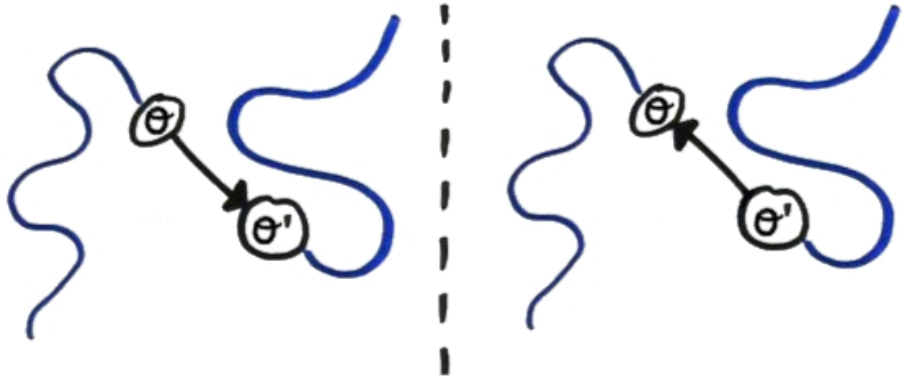
- A **sufficient** condition used for constructing valid Markov chains is to ensure that the chain satisfies **detailed balance**:

$$T(\theta' \leftarrow \theta) \cdot p(\theta|\mathcal{D}) = T(\theta \leftarrow \theta') \cdot p(\theta'|\mathcal{D})$$

  Chains that satisfy detailed balance are known as **reversible**

- Detailed balance $\Longrightarrow$ Stationarity

- Hence, construct MCMC samplers by using detailed balance to construct a valid transition kernel



* Image Credit: Iain Murray

Metropolis Hastings (MH)

- MH is one of the simplest and most widely used MCMC methods
- Given an unnormalized target $p(\theta|\mathcal{D})$ (hereafter $p(\theta) \equiv p(\theta, \mathcal{D})$ for simplicity), a starting point θ_1, and a proposal $q(\theta'|\theta)$, the MH algorithm repeatedly applies the following steps ad infinitum
  1. Propose a new point $\theta' \sim q(\theta'|\theta)$ (where θ is the sample in the previous time step)
  2. Accept the new sample θ' with probability

$$P_{\text{accept}} = \min\left\{1, \frac{p(\theta')\cdot q(\theta|\theta')}{p(\theta)\cdot q(\theta'|\theta)}\right\}$$

  3. If the new sample is rejected, accept the previous sample θ' (i.e., repeat θ' in this time step)
  4. Go back to 1

Metropolis Hastings (MH) (2)

- For example, we can choose a symmetric proposal $q(\theta'|\theta)$, such that

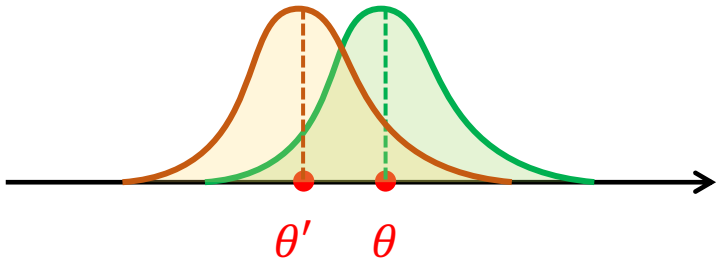
$$q(\theta|\theta') = q(\theta'|\theta)$$

(e.g., let $q(\theta'|\theta) = \mathcal{N}(\theta, 1)$). We can simplify the acceptance probability as:

$$P_{\text{accept}} = \min\left\{1, \frac{p(\theta')}{p(\theta)}\right\}$$

- Intuitively, always accept θ' if $p(\theta') \geq p(\theta)$.

Otherwise, accept θ' with probability $\frac{p(\theta')}{p(\theta)}$



symmetric proposal

Metropolis Hastings (MH) (2)

- For example, we can choose a symmetric proposal $q(\theta'|\theta)$, such that

$$q(\theta|\theta') = q(\theta'|\theta)$$

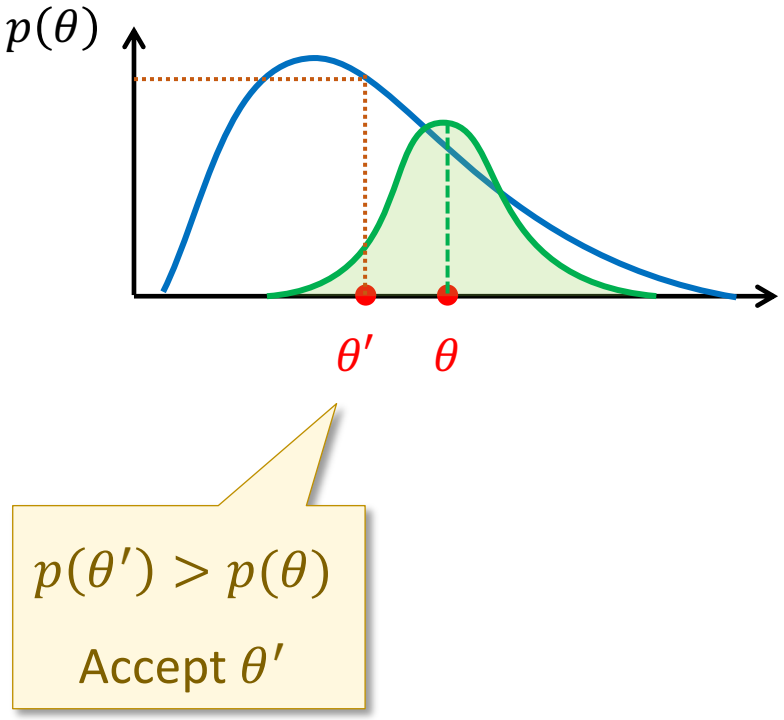
(e.g., let $q(\theta'|\theta) = \mathcal{N}(\theta, 1)$). We can simplify the acceptance probability as:

$$P_{\text{accept}} = \min\left\{1, \frac{p(\theta')}{p(\theta)}\right\}$$

- Intuitively, always accept θ' if $p(\theta') \geq p(\theta)$.

  Otherwise, accept θ' with probability $\frac{p(\theta')}{p(\theta)}$

  - Hill–climbing effect



Metropolis Hastings (MH) (2)

- For example, we can choose a symmetric proposal $q(\theta'|\theta)$, such that

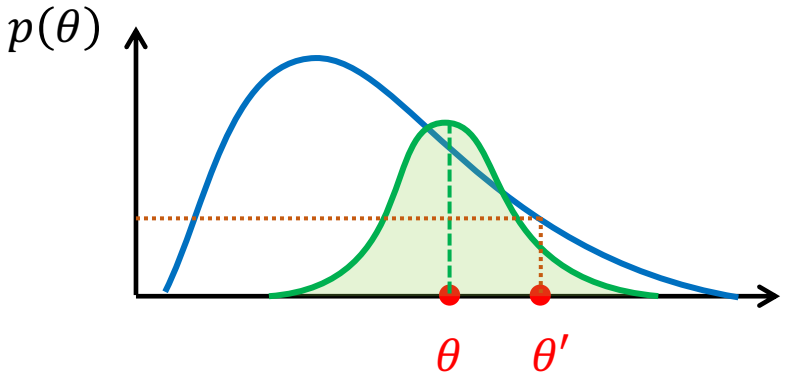
$$q(\theta|\theta') = q(\theta'|\theta)$$

(e.g., let $q(\theta'|\theta) = \mathcal{N}(\theta, 1)$). We can simplify the acceptance probability as:

$$P_{\text{accept}} = \min\left\{1, \frac{p(\theta')}{p(\theta)}\right\}$$

- Intuitively, always accept θ' if $p(\theta') \geq p(\theta)$.

Otherwise, accept θ' with probability $\frac{p(\theta')}{p(\theta)}$



$p(\theta') < p(\theta)$
Accept θ' with probability $p(\theta')/p(\theta)$

Metropolis Hastings (MH) (2)

- For example, we can choose a symmetric proposal $q(\theta'|\theta)$, such that

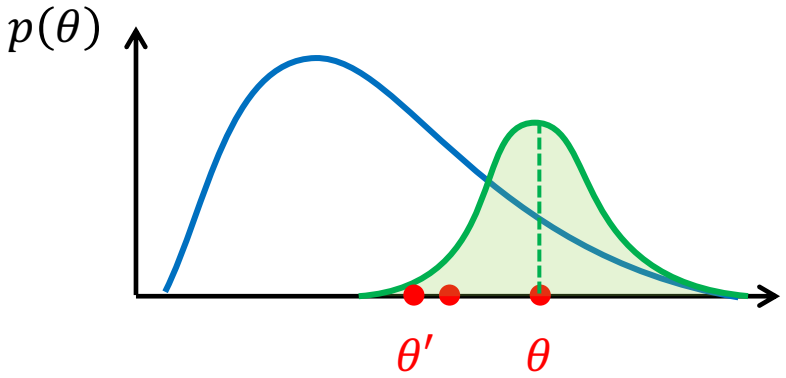
$$q(\theta|\theta') = q(\theta'|\theta)$$

(e.g., let $q(\theta'|\theta) = \mathcal{N}(\theta, 1)$). We can simplify the acceptance probability as:

$$P_{\text{accept}} = \min\left\{1, \frac{p(\theta')}{p(\theta)}\right\}$$

- Intuitively, always accept θ' if $p(\theta') \geq p(\theta)$.

Otherwise, accept θ' with probability $\frac{p(\theta')}{p(\theta)}$



Metropolis Hastings (MH) (2)

- For example, we can choose a symmetric proposal $q(\theta'|\theta)$, such that

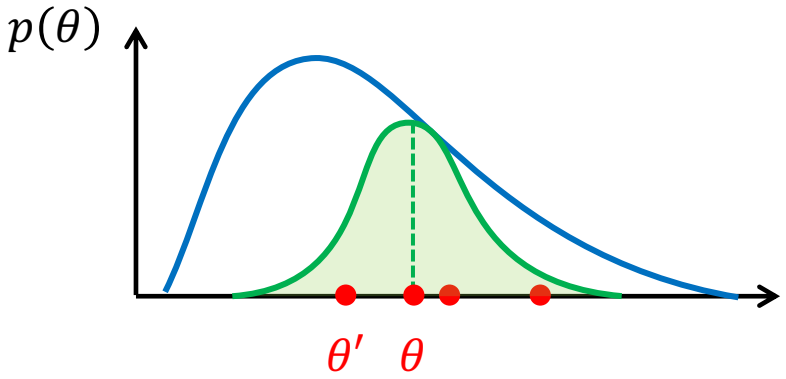
$$q(\theta|\theta') = q(\theta'|\theta)$$

(e.g., let $q(\theta'|\theta) = \mathcal{N}(\theta, 1)$). We can simplify the acceptance probability as:

$$P_{\text{accept}} = \min\left\{1, \frac{p(\theta')}{p(\theta)}\right\}$$

- Intuitively, always accept θ' if $p(\theta') \geq p(\theta)$.

Otherwise, accept θ' with probability $\frac{p(\theta')}{p(\theta)}$



Metropolis Hastings (MH) (2)

- For example, we can choose a symmetric proposal $q(\theta'|\theta)$, such that

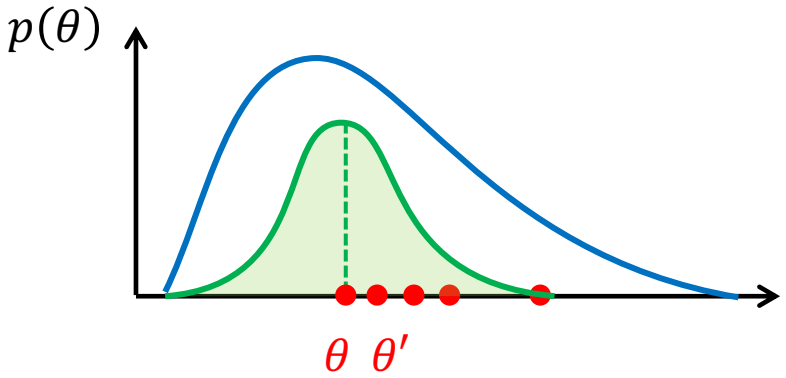
$$q(\theta|\theta') = q(\theta'|\theta)$$

(e.g., let $q(\theta'|\theta) = \mathcal{N}(\theta, 1)$). We can simplify the acceptance probability as:

$$P_{\text{accept}} = \min\left\{1, \frac{p(\theta')}{p(\theta)}\right\}$$

- Intuitively, always accept θ' if $p(\theta') \geq p(\theta)$.

Otherwise, accept θ' with probability $\frac{p(\theta')}{p(\theta)}$



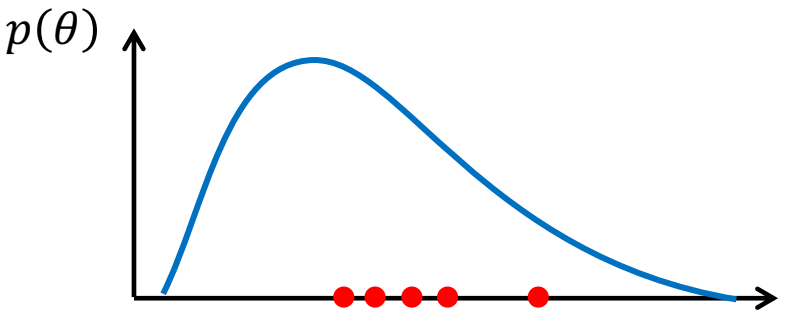
Metropolis Hastings (MH) (3)

- This produces an infinite sequence of samples $\theta_1, \theta_2, \ldots, \theta_n, \ldots$ that converge to $p(\theta)$ and from which we can construct a **Monte Carlo estimator**

$$p(\theta) \approx \frac{1}{N} \sum_{n=M}^{N} \mathbb{I}(\Theta_n = \theta)$$

  where we start with sample M to burn-in the chain

- Unlike rejection/importance sampling, the **samples are correlated** and produce **biased estimates** for finite N
- The key though is that the proposal $q(\theta'|\theta)$ depends on the current position allowing us to make **local moves**



An infinite sequence of samples can be generated

☞ **MCMC Demo:** https://chi-feng.github.io/mcmc-demo/app.html?algorithm=RandomWalkMH&target=banana

More Advanced MCMC Methods

- There are loads of more advanced MCMC methods.
- Two that are particularly prominent ones that you should be able to quickly pick up given what you have already learned are:
  - Gibbs sampling (see the notes)
  - Hamiltonian Monte Carlo: https://arxiv.org/pdf/1206.1901.pdf?fname=cm&font=TypeI

    Demo: https://chi-feng.github.io/mcmc-demo/app.html?algorithm=HamiltonianMC&target=donut

Pros and Cons of MCMC Methods

Pros

- Able to work in **high dimensions** due to making local moves
- No requirement to have normalized target
- Consistent in the limit of running the chain for an infinitely long time
- Do not require as finely tuned proposals as importance sampling or rejection sampling
- Surprisingly effective for a huge range of problems

Cons

- Produce **biased estimates** for finite sample sizes due to correlation between samples
- Diagnostics can be very difficult
- Typically struggle to deal with **multiple modes**
- Proposal still quite important: chain can mix very slowly if the proposal is not good
- Can be **difficult to parallelize**
- Deriving theoretical results is more difficult than previous approaches
- Produces no marginal likelihood estimate
- Typically far slower to converge than the variational methods we introduce next

Variational Inference

Variational Inference (VI)

- Another class of ubiquitously used approaches for Bayesian inference wherein we try to learn an approximation to $p(\theta|\mathcal{D})$
- Key idea: reformulate the inference problem to an **optimization**, find a best distribution to approximate $p(\theta|\mathcal{D})$ from a set of candidate distributions
  - The candidate distributions are from a **parameterized variational family** $q_\varphi(\theta), \varphi \in \Phi$. (For example, φ are the weights in a neural network.)
  - Then finding the $\varphi^* \in \Phi$ that gives the "best" approximation based on the **Kullback–Leibler (KL) divergence** $\mathrm{KL}(q \parallel p)$:

$$\varphi^* = \mathrm{argmin}_{\varphi \in \Phi} \mathrm{KL}\left(q_\varphi(\theta) \parallel p(\theta|\mathcal{D})\right)$$

KL Divergence

- The KL divergence measures how similar two distributions $p(x)$ and $q(x)$ are to one another (intuitively, the distance between them). It is defined as

$$\mathrm{KL}(q \parallel p) = \int q(x) \cdot \log \frac{q(x)}{p(x)} \, dx = \mathbb{E}_{x \sim q(x)} \left[\log \frac{q(x)}{p(x)} \right]$$

- Important properties:
  - $\mathrm{KL}(q \parallel p) \geq 0$ for any p and q
  - $\mathrm{KL}(q \parallel p) = 0$ if and only if $p(x) = q(x)$ for all x
  - In general, $\mathrm{KL}(q \parallel p) \neq \mathrm{KL}(p \parallel q)$

Variational Inference (VI)

- We cannot work directly with $\mathrm{KL}(q_\varphi(\theta) \parallel p(\theta|\mathcal{D}))$ because we don't know the posterior density
- We note that the marginal likelihood $p(\mathcal{D})$ is independent of our variational parameters φ to work with the joint instead (see the right)
- We work with $\mathrm{KL}(q_\varphi(\theta) \parallel p(\theta|\mathcal{D}))$ rather than $\mathrm{KL}\left(p(\theta|\mathcal{D}) \parallel q_\varphi(\theta)\right)$ because the latter is doubly intractable

$$
\begin{aligned}
\phi^* &= \arg\min_{\phi\in\varphi} \mathrm{KL}(q_\phi(\theta) \parallel p(\theta|\mathcal{D})) \\
&= \arg\min_{\phi\in\varphi} \mathbb{E}_{q_\phi(\theta)}\left[\log \frac{q_\phi(\theta)}{p(\theta|\mathcal{D})}\right] \\
&= \arg\min_{\phi\in\varphi} \mathbb{E}_{q_\phi(\theta)}\left[\log \frac{q_\phi(\theta)}{p(\theta|\mathcal{D})}\right] - \log p(\mathcal{D}) \\
&= \arg\min_{\phi\in\varphi} \mathbb{E}_{q_\phi(\theta)}\left[\log \frac{q_\phi(\theta)}{p(\theta,\mathcal{D})}\right]
\end{aligned}
$$

The ELBO (1)

- We can equivalently think about the optimization problem in VI as the maximization

$$\varphi^* = \arg\max_{\varphi \in \Phi} \mathcal{L}(\varphi),$$

where

$$\begin{aligned}\mathcal{L}(\varphi) &:= \mathbb{E}_{\theta \sim q_\varphi}\left[\log \frac{p(\theta, \mathcal{D})}{q_\varphi(\theta)}\right] \\ &= \log p(\mathcal{D}) - \mathrm{KL}\left(q_\varphi \parallel p(\cdot \mid \mathcal{D})\right)\end{aligned}$$

is known as the **Evidence Lower BOund (ELBO)**. $\mathcal{L}(\varphi)$ is a lower bound on the log evidence, i.e., we have $\mathcal{L}(\varphi) \geq \log p(\mathcal{D})$. It is also sometimes known as the variational free energy

Example: Gaussian with Unknown Mean and Variance

As a simple worked example (taken from Bishop 10.1.3), consider the following model where we are trying to infer to the mean μ and precision τ of a Gaussian given a set of observations $\mathcal{D} = \{x_n\}_{n=1}^N$.

Our full model is given by

$$p(\tau) = \text{GAMMA}(\tau; \alpha, \beta)$$
$$p(\mu|\tau) = \mathcal{N}(\mu; \mu_0, (\lambda_0\tau)^{-1})$$
$$p(\mathcal{D}|\underbrace{\mu, \tau}_{\theta}) = \prod_{n=1}^{N} \mathcal{N}(x_n; \mu, \tau^{-1})$$

Example: Gaussian with Unknown Mean and Variance

We care about the posterior $p(\mu, \tau | \mathcal{D})$ and we are going to try and approximate this using variational inference

For our variational family we will take

$$
\begin{aligned}
q_\phi(\tau, \mu) &= q(\tau) q(\mu) \\
q_\phi(\tau) &= \textsc{Gamma}(\tau; \phi_a, \phi_b) \\
q_\phi(\mu) &= \mathcal{N}(\mu; \phi_c, \phi_d^{-1})
\end{aligned}
$$

where we note that this factorization is an assumption: the posterior itself does not factorize

Example: Gaussian with Unknown Mean and Variance

To find the best variational parameters ϕ^*, we need to optimize $\mathcal{L}(\phi)$, for which we can use gradient methods, using

$$\nabla_\phi \mathcal{L}(\phi) = \nabla_\phi \iint q_\phi(\tau) q_\phi(\mu) \log \left(\frac{p(\mathcal{D}|\mu,\tau)p(\mu|\tau)p(\tau)}{q_\phi(\tau)q_\phi(\mu)} \right) d\tau d\mu$$

If we can calculate this gradient, this means we can optimize ϕ by performing gradient ascent.

After initializing some ϕ_0, we just repeatedly apply

$$\phi_{n+1} \leftarrow \phi_n + \epsilon_n \nabla_\phi \mathcal{L}(\phi_n)$$

where ϵ_n are our step sizes

Example: Gaussian with Unknown Mean and Variance

To find the best variational parameters ϕ^*, we need to optimize $\mathcal{L}(\phi)$, for which we can use gradient methods, using

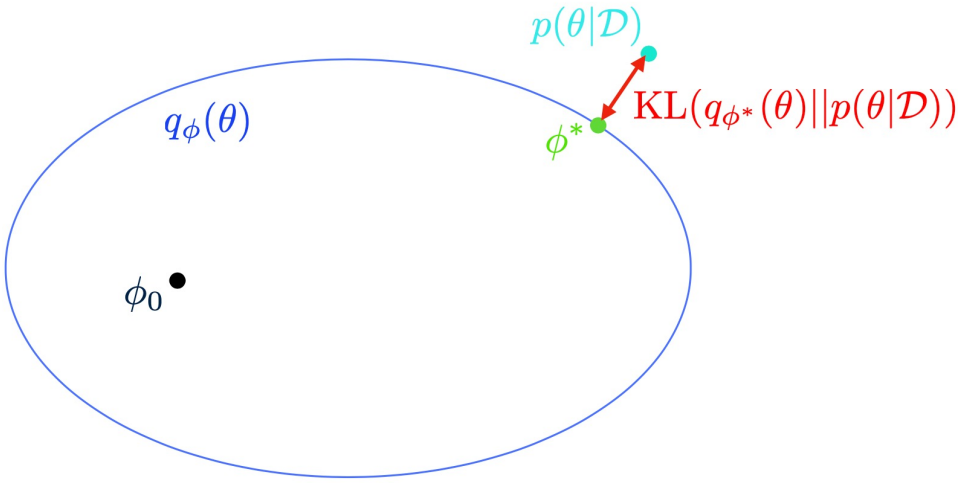
$$\nabla_\phi \mathcal{L}(\phi) = \nabla_\phi \iint q_\phi(\tau) q_\phi(\mu) \log \left(\frac{p(\mathcal{D}|\mu,\tau)p(\mu|\tau)p(\tau)}{q_\phi(\tau)q_\phi(\mu)} \right) d\tau d\mu$$

If we can calculate this gradient, this means we can optimize ϕ by performing gradient ascent.

After initializing some ϕ_0, we just repeatedly apply

$$\phi_{n+1} \leftarrow \phi_n + \epsilon_n \nabla_\phi \mathcal{L}(\phi_n)$$

where ϵ_n are our step sizes



Example: Gaussian with Unknown Mean and Variance

To find the best variational parameters ϕ^*, we need to optimize $\mathcal{L}(\phi)$, for which we can use gradient methods, using

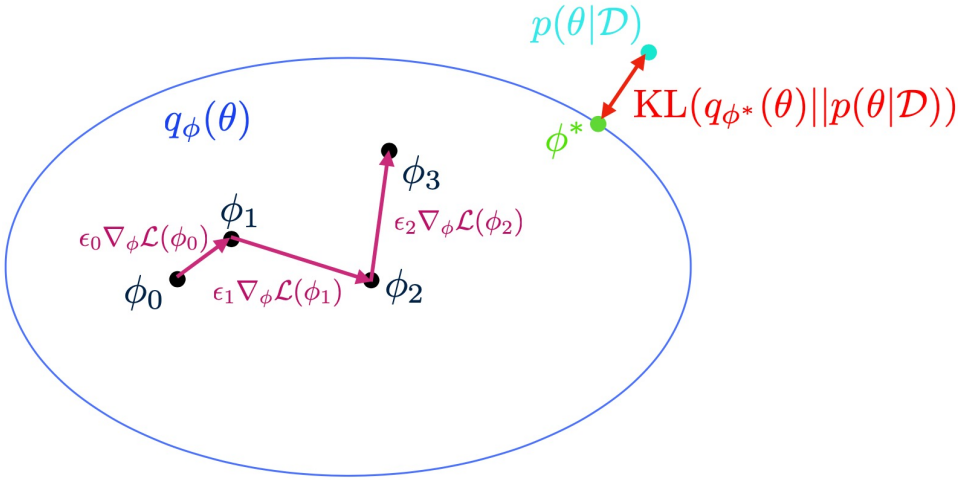
$$\nabla_\phi \mathcal{L}(\phi) = \nabla_\phi \iint q_\phi(\tau) q_\phi(\mu) \log \left(\frac{p(\mathcal{D}|\mu,\tau)p(\mu|\tau)p(\tau)}{q_\phi(\tau)q_\phi(\mu)} \right) d\tau d\mu$$

If we can calculate this gradient, this means we can optimize ϕ by performing gradient ascent.

After initializing some ϕ_0, we just repeatedly apply

$$\phi_{n+1} \leftarrow \phi_n + \epsilon_n \nabla_\phi \mathcal{L}(\phi_n)$$

where ϵ_n are our step sizes



Example: Gaussian with Unknown Mean and Variance

To find the best variational parameters ϕ^*, we need to optimize $\mathcal{L}(\phi)$, for which we can use gradient methods, using

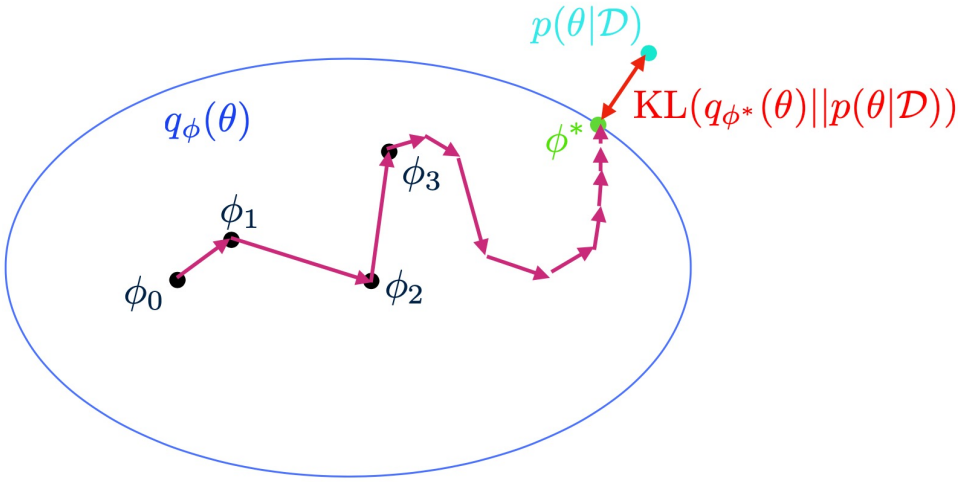
$$\nabla_\phi \mathcal{L}(\phi) = \nabla_\phi \iint q_\phi(\tau) q_\phi(\mu) \log \left(\frac{p(\mathcal{D}|\mu,\tau)p(\mu|\tau)p(\tau)}{q_\phi(\tau)q_\phi(\mu)} \right) d\tau d\mu$$

If we can calculate this gradient, this means we can optimize ϕ by performing gradient ascent.

After initializing some ϕ_0, we just repeatedly apply

$$\phi_{n+1} \leftarrow \phi_n + \epsilon_n \nabla_\phi \mathcal{L}(\phi_n)$$

where ϵ_n are our step sizes



Pros and Cons of Variational Methods

Pros

- Typically **more efficient** than MCMC approaches, particularly in high dimensions once we exploit the stochastic variational approaches introduced in the next lecture
- Can often provided effective inference for models where MCMC methods have impractically slow convergence
- Though it is an approximation for the density, we can also sample directly from our variational distribution to calculate Monte Carlo estimates if needed
- Allows simultaneous optimization of model parameters

Cons

- But it produces (potentially very) **biased estimates** and requires **strong structural assumptions** to be made about the form of the posterior
- Unlike MCMC methods, this bias stays even in the limit of large computation
- Often requires substantial tailoring to a particular problem
- Very difficult to estimate how much error there is in the approximation: subsequent estimates can be unreliable, particular in their uncertainty
- Tends to underestimate the variance of the posterior due to mode–seeking nature of reverse KL, particularly if using a mean field assumption

Variational Auto–Encoders

Variational Auto–Encoders (VAEs) (1)

We can also view the VAE as a stochastic auto–encoder where the inference network=encoder and the generative network=decoder:

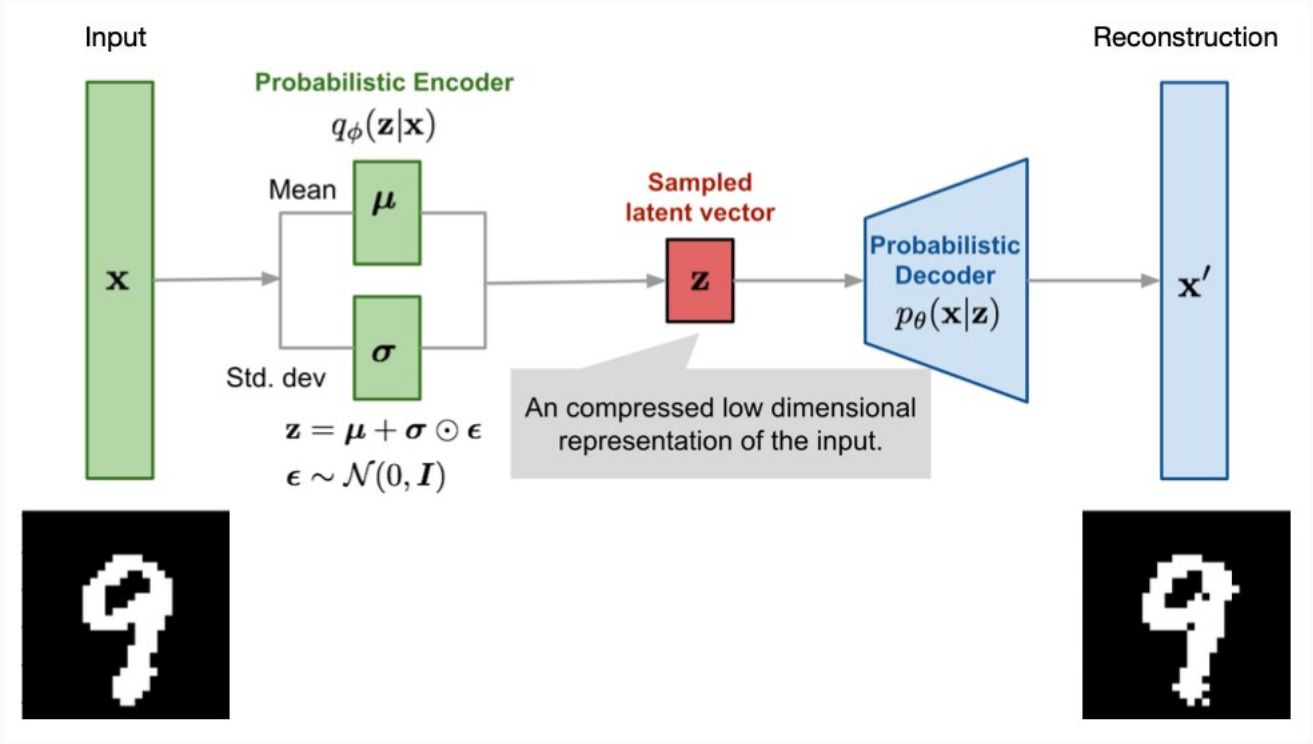
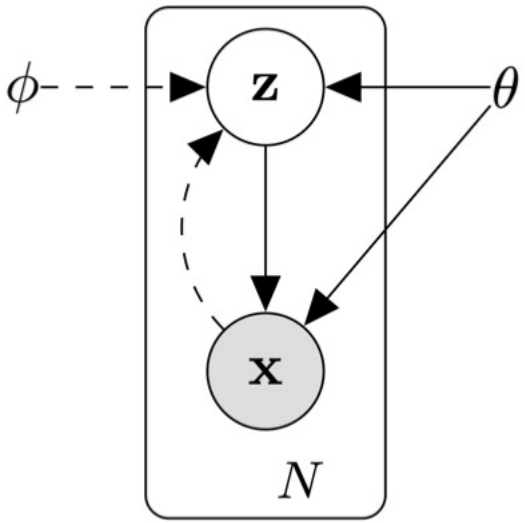


Image adapted from one by Lilian Weng

Variational Auto–Encoders (VAEs) (2)



$q_\phi(z|x)$

Inference Network

$p_\theta(x|z)$

Generative Network

Maximize

$$\mathcal{L}(\theta, \phi, \mathcal{D}) := \sum_{n=1}^{N} \mathbb{E}_{q_\phi(z_n|x_n)} \left[\log \left(\frac{p_\theta(x_n, z_n)}{q_\phi(z_n|x_n)} \right) \right] \quad (7)$$

with respect to both θ and ϕ

Further Reading (1)

- The lecture notes give extra information on the curse of dimensionality and MCMC methods
- Iain Murray on MCMC https://www.youtube.com/watch?v=_v4Eb09qp7Q
- Chapters 21, 22, and 23 of K P Murphy. Machine learning: a probabilistic perspective. 2012
- David M Blei, Alp Kucukelbir, and Jon D McAuliffe. “Variational inference: A review for statisticians”. In: Journal of the American statistical Association (2017)
- NeurIPS tutorial on variational inference that accompanies the previous paper: https://www.youtube.com/watch?v=ogdv_6dbvVQ

Further Reading (2)

- The are no additional lecture notes for this lecture: you need to go investigate for yourself
- Training VAEs in Pyro: https://pyro.ai/examples/vae.html and https://www.youtube.com/watch?v=vgFWeEyen6Y&t=1058s
- Tutorial paper on VAEs: Carl Doersch. “Tutorial on variational autoencoders”. In: arXiv preprint arXiv:1606.05908 (2016)
- Video tutorial on deep generative models by Shakir Mohamed and Danilo Rezende https://www.youtube.com/watch?v=JrO5fSskISY
- GANs, one of the main alternatives to VAEs: Ian Goodfellow et al. “Generative adversarial nets”. In: Advances in neural information processing systems. 2014